PL-TR-96-2024

PHASE 2 DEVELOPMENT TASKS: RESEARCH TO ENHANCE THE PHILLIPS LABORATORY SIMLAB CAPABILITY FOR SIMULATING THE IR SCENE

Keith Johnson
Allen Curran
Eric Marttila
Ban Yee
Alan Koivunen
Leonard Rodriguez

Michigan Technological University
1400 Townsend Drive
Houghton, Michigan 49931-1295

10 January 1996

Final Report
June 1993 - December 1995

Approved for public release; distribution unlimited.

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AFB, MA 01731-3010

"This technical report has been reviewed and is approved for publication"

JOSEPH P. ALLECA, Lt Col, USAF
Contract Manager

DONALD D. GRANTHAM, Chief
Atmospheric Structure Branch

Arnold A. Barnes, Jr., Acting
ROBERT A. McCLATCHEY, Director
Atmospheric Sciences Division

This report has been reviewed by the ESC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center (DTIC). All others should apply to the National Technical Information Service (NTIS).

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/TSI, 29 Randoph Road, Hanscom AFB, MA. 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations of notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10 January 1996	Final June 1993 - December 1995

4. TITLE AND SUBTITLE
Phase 2 Development Tasks: Research to Enhance the Phillips Laboratory SIMLAB Capability for Simulating the IR Scene

5. FUNDING NUMBERS
PE 63707F
PR 2688 TA GU WU MA
Contract F19628-93-K-0020

6. AUTHOR(S)
Keith Johnson
Allen Curran
Eric Marttila
Ban Yee
Alan Koivunen
Leonard Rodriguez

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Michigan Technological University
1400 Townsend Drive
Houghton, MI 49931-1295

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Phillips Laboratory
29 Randolph Road
Hanscom AFB, MA 01731-3010

Contract Manager: J. Alleca, Lt Col, USAF/GPAA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
PL-TR-96-2024

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Thermal Contrast Model (TCM2) was originally written in Fortran. Phase 1 of this contract resulted in a conversion to the Mathematica language for research purposes. Phase 2 has concentrated on a future operational implementation in ANSI for use with the Phillips Lab Scene Simulation software.

This document includes a TCM2.C (ANSI C) target model inventory, description of the TCM2.C file formats, and theoretical discussions of the sloped background and BRDF algorithms.

14. SUBJECT TERMS
Thermal Contrast Model, Tactical Decision Aid, Bidirectional Reflectance Distribution Function, IR Targets

15. NUMBER OF PAGES
58

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Table of Contents

Preface

This work was performed under contract number F19628-93-K0020 for the Geophysics Directorate of Phillips Laboratory at Hanscom AFB, MA. The program was divided into two phases covering a 30 month period (June 23, 1993 - December 31, 1995). The first phase was the development of a Mathematica Thermal Contrast Model which was delivered and reported on Sept 21, 1994. This final phase covers the activities for the period from September 1994 to December 31, 1995.

The work was performed by the Applied Research Group of the Keweenaw Research Center of Michigan Technological University. The program manager was Keith R. Johnson and the technical contributors were Dr. Allen R. Curran, Eric A. Marttila, Ban K. Yee, Alan C. Koivunen, and Dr. Leonard J. Rodriguez.

The contract was under the direction of Lt. Col. Joseph Alleca (previously Capt. Joseph Eicher) of PL/GPAA.

1. INTRODUCTION

The Thermal Contrast Model (TCM2) was originally written in Fortran[1]. Phase 1 of this contract resulted in a conversion to the Mathematica language for research purposes[2]. Phase 2 has concentrated on a future operational implementation in ANSI C for use with the Phillips Lab Scene Simulation software.

This document includes a TCM2.C (ANSI C) target model inventory, description of the TCM2.C file formats, and theoretical discussions of the sloped background and BRDF algorithms.

[1] Johnson and Rodriguez, "User's Manual for TCM2," Three Volume Georgia Tech Report, Air Force Contract No. F33615-88-C-1865, January 1991.

[2] Marttila, Curran, Yee, and Johnson, "Using The MxCDA Thermal Model - Phase 1," September 21, 1994.

2. TCM2.C TARGET INVENTORY

The following target inventory represents all of the models that have been converted and upgraded from the original Research Grade Tactical Decision Aid (RGTDA) model database[3] (and other Fortran TCM2/GTSIG models) for use with TCM2.C.

Table 1: TCM2.C Target Inventory

Target Model	Type	Comments
AH64	ROTARY WING VEHICLE	Apache Helicopter 2 Modes: Hovering and Off
AIRFIELD	HIGH VALUE TARGET	Runways & Buildings: 3 States: Dry, Normal, Wet
BMP2	GROUND VEHICLE	Russian 3 Modes: Exercised, Idling, Off
BRADLEY (M2)	GROUND VEHICLE	US 3 Modes: Exercised, Idling, Off
BRDM	GROUND VEHICLE	Russian 3 Modes: Exercised, Idling, Off
BRIDGE	HIGH VALUE TARGET	Highway Bridge
BUILDING	HIGH VALUE TARGET	3 Buildings: 1. One story frame building with hip shingle roof and controlled temp interior. 2. Two story masonry bldg with flat shingle roof and controlled temp interior. 3. "Tin" storage shed with flat "tin" roof and no ventilation.
BUNKER	HIGH VALUE TARGET	Aircraft Bunker
CAMO NETS	STATIONARY GROUND TARGET	6 Versions: Snow (Full and Partial Snow Coverage; Woodland (Spring/Summer and Fall/Winter); Desert (Arid and Semi-Arid).
DAM	HIGH VALUE TARGET	Lock and Dam
F4	FIXED WING VEHICLE	Parked (Off) 2 Modes: Full fuel tank and empty fuel tank.

[3] Johnson and Rodriguez, "Target/Background Models for GTSIG/TCM2: Volume 3," GTRI, 1/92.

Target Model	Type	Comments
GENERATOR	STATIONARY GROUND TARGET	Small Diesel Generator 2 Modes: Idling and Off.
HIND	ROTARY WING VEHICLE	Russian Mi-24 Helicopter 2 Modes: Hovering and Off
HR64	ROTARY WING VEHICLE	Apache Helicopter (Hover Mode) High Resolution Version
KRIVAK	SHIP	Russian Missile Frigate Ship: Exercised
PAVELOW	ROTARY WING VEHICLE	MH53-J US Helicopter Hovering
PWRPLANT	HIGH VALUE TARGET	Hydroelectric Powerplant Operating
REFINERY	HIGH VALUE TARGET	Oil Refinery Operating
SA12	STATIONARY GROUND TARGET	3 Vehicles: C^3 on Vehicle: Exer, Idle, Off Missiles and Launcher Radar Unit on Vehicle
SCUD	STATIONARY GROUND TARGET	Russian Scud Missile and Launcher 3 Modes: Exercised, Idling, Off
SHIP	SHIP	Small Research Vessel Operating
T62	GROUND VEHICLE	Russian Tank 3 Modes: Exercised, Idling, Off
T72	GROUND VEHICLE	Russian Tank 3 Modes: Exercised, Idling, Off
T80	GROUND VEHICLE	Russian Tank 3 Modes: Exercised, Idling, Off
TANK	GROUND VEHICLE	Similar to M1 3 Modes: Exercised, Idling, Off
ZIL	GROUND VEHICLE	Russian Truck 3 Modes: Exercised, Idling, Off

3. THERMAL MODEL INPUT AND OUTPUT FILES

The following list of input and output files are used in TCM2.

Input Files:

scene description file (scene.sdf): Describes the scene setting, time, date, targets and backgrounds.

nodal parameter file (model.par): Created by the thermal modeler. Contains nodal network information for the target.

radiation exchange file (model.lwx): Created by the thermal modeler. Contains multi-bounce radiation exchange factors for the target's thermal nodal network.

apparent area file (model.apa): Created by the thermal modeler. Contains target apparent areas from different view points. Used for solar loading and contrast history calculations.

facet file (model.fac): Created by the thermal modeler. Contains target geometry information.

weather file (name.nc): Contains weather data for the simulation time.

numerics file (numeric.prm): Contains information for the thermal model numerical solving routines (convergence criteria, relaxation factors, etc.).

options file (no naming convention): A text file whose use is limited to model development and verification to activate additional output and non-default processing.

air property file (airprop.dat): Contains air property data as a function of temperature.

water property file (h2oprop.dat): Contains water property data as a function of temperature.

Output Files Created by TCM2:

scene radiance file (scene.srf): Contains scene radiances for all targets and backgrounds. This file is read by the PL Scene Builder/Viewer and used for "Thermal Shading".

apparent temperature file: (model.app): Contain target and primary background apparent temperatures for the entire simulation.

physical temperature file: (model.pht): Contain target and primary background physical temperatures for the entire simulation. Data in these files can be compared with thermocouple data during model assessment.

log file (tcm2.log): Log file of the last thermal model run. This file contains a running log of some intermediate calculations that may be of use to the user. The file also contains statements that indicate the status of the thermal model run.

3.1 Description of the Scene Description File (SDF)

The primary file that the user will control the scenario inputs is with the SDF File. This is an ASCII file that can be created and edited with a conventional text editor. The input parameters are listed below

The SDF File Format: The Scene Description File consists of the following 3 record types:

File Header:	latitude, longitude, elevation, year, month, day, endTime, iTime, minSensorWavelength, maxSensorWavelength, nBg, nTarget
Background Record:	bgID, bgType, Normal[3]
Target Record:	ID, bearing, speed, bgID[4]

where

latitude:	real - mean latitude of the scene (deg North)
longitude:	real - mean longitude of the scene (deg West)
elevation:	real - mean elevation of the scene (meters)
year:	integer - year portion of scene date (4 digits)
month:	integer - month portion of scene date (1 to 12)
day:	integer - day portion of scene date (1 to 31)
endTime:	integer - final scene time (GMT: HHMMSS)
iTime:	integer - hours prior to scene time to begin simulation (HHMMSS)
minSensorWavelength:	real (microns)
maxSensorWavelength:	real (microns)
nBg:	integer - number of Background Records to follow the header
nTarget:	integer - number of Target Records following Background Records
bgID:	integer - background ID
bgType:	integer - background type indicator
Normal:	real - unit normal of background (North East Up)
ID:	string - unique target-model identifier
bearing:	real - rotation of target axis from North (deg, positive towards east)
speed:	real - instantaneous speed of target along its bearing (m/s)
bgID:	backward reference to a Background Record

File Organization: The file organization is illustrated below.

File Header
- Background Record 1
- Background Record 2
- .
- .
- .
- Background Record nBg
- Target Record 1
- Target Record 2
- .
- .
- .
- Target Record nTarget

Table 2: Background Choices and Parameter Options

BgType	Name	Current Key	Parameters
0	*Unknown*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
1	*Deciduous*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
2	*Coniferous*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
3	*Mixed Deciduous/Coniferous*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
4	*Grass, tall*	**FOLIAGE**	`{0.0,1.0,-1,-1,-1}`
5	*Grass, short*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
6	*Sand*	**SOIL**	`{7,-1,-1,-1,-1}`
7	*Soil*	**SOIL**	`{-1,-1,-1,-1,-1}`
8	*Concrete*	**SLAB_CONCRETE**	`{-1,-1,-1,-1,-1}`
9	*Asphalt*	**SLAB_ASPHALT**	`{-1,-1,-1,-1,-1}`
10	*Water*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
11	*Residential*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`
12	*Commercial*	**FOLIAGE**	`{0.0,0.5,-1,-1,-1}`

Background Adjustment Parameters:

BKG_ID = SOIL

Par 1 = STYPE (Soil Type)

1 Average Soil (D)
2 Loam
3 Sandy Soil
4 Clay
5 Peat
6 Gravel
7 Desert Sand

Par 2 = N/A (USE ANY DUMMY VALUE)

Par 3 = ABSIN (Solar Absorptivity of Surface - (DIMENSIONLESS) [0. - 1.] or a -1 to utilize internal model value dependent on soil moisture)

-1 Model Value (D)

Par 4 = WGIN (INITIAL VOLUME CONCENTRATION OF SURFACE DEPTH MOISTURE - (DIMENSIONLESS) [0. - 1.])

0.0 DRY (Default for Desert only)
0.2 MODERATE (Default for all others)
0.3 - 1.0 SATURATED

Par 5 = W2IN (INITIAL VOL. CONCENTRATION OF BULK DEPTH MOISTURE (DIMENSIONLESS) [0. - 1.])

0.0 DRY (Default for Desert only)
0.25 MODERATE (Default for all others)
0.3 - 1.0 SATURATED

BKG_ID = WATER {Future Implementation}

Par 1 = WTYPE (Water Type)

1 First Principles (D)
2 Empirical
3 Constant (TCORE)

Par 2 = DEEP (TOTAL DEPTH - METERS) **[DEFAULT = 5 M]**

Par 3 = ETA (SOLAR ATTENUATION OF WATER [0.05 - 1.0])

0.05 CLEAR WATER (D)
1.0 TURBID WATER

Par 4 = ABSIN (Solar Absorptivity of Surface - (DIMENSIONLESS) [0. - 1.] or a -1 to utilize internal model value dependent on soil moisture)

-1 Model Value (D)

Par 5 = N/A (USE A -1 VALUE)

BKG_ID = SNOW {Future Implementation}

Par 1=SNOTYP (SNOW TYPE [11 - 14] or Solar Abs [0.0 - 1.0])
0.0 - 1.0 User Input of Solar Absorptivity
11 Fresh (D)
12 Old Dry
13 Rained Upon
14 Surface Melted

Par 2=DAZML (NUMBER OF DAYS OF MELTING) **[DEFAULT = 0.]**

Par 3=SNODEP (SNOW DEPTH - METERS) **[DEFAULT = 1.]**

Par 4=SNOCON (SNOW CONDITION)
1 Compacted By Vehicles
2 Windy Region
3 Late In Season
4 Tundra
5 Undisturbed (D)

Par 5=N/A (USE ANY DUMMY VALUE)

BKG_ID = FOLIAGE

Par 1 = ZIG (GROWING CONDITION FACTOR [0. - 1000.])
0.0 GROWING (D)
5.0 INTERMEDIATE
1000. DORMANT

Par 2 = SIGF (FOLIAGE COVER FACTOR [0. - 1.])
0.0 NO FOLIAGE
.75 INTERMEDIATE (D))
1.0 COMPLETE COVERAGE

Par 3 = ABSIN (Solar Absorptivity of Surface - (DIMENSIONLESS) [0. - 1.] or a -1 to utilize internal model value dependent on soil moisture)
-1 Model Value (D)

Par 4 = WGIN (INITIAL VOLUME CONCENTRATION OF SURFACE DEPTH MOISTURE - (DIMENSIONLESS) [0. - 1.])
0.0 DRY
0.2 MODERATE (D)
0.3 - 1.0 SATURATED

Par 5 = W2IN (INITIAL VOL. CONCENTRATION OF BULK DEPTH MOISTURE (DIMENSIONLESS) [0. - 1.])
0. DRY
.25 MODERATE (D)
0.3 - 1.0 SATURATED

BKG_ID = SLAB_CONCRETE

Par 1 = ITYPE (STRUCTURAL CONSTRUCTION TYPE [1 - 6])
1 INTERSTATE ROAD
2 SIDEWALK
3 RUNWAY (D)
4 PARKING LOT
5 HIGHWAY BRIDGE
6 HEAVY PAD

Par 2 = ISURF (SURFACE CONDITION [1 - 3] or Solar Abs.[0.0 - 0.99])
1 UNCOLORED (D)
2 BLACK
3 BROWN

Par 3 = IWET (WETNESS CONDITION [1 - 3])
1 COVERED (DRY)
2 EXPOSED (NORMAL) (D)
3 WET

Par 4 = N/A (USE ANY DUMMY VALUE)
Par 5 = N/A (USE ANY DUMMY VALUE)

BKG_ID = SLAB_ASPHALT

Par 1 = ITYPE (STRUCTURAL CONSTRUCTION TYPE [1 - 5])
1 INTERSTATE ROAD
2 COUNTRY ROAD
3 RUNWAY (D)
4 PARKING LOT
5 HIGHWAY BRIDGE

Par 2 = ISURF (SURFACE CONDITION [1 - 2] or Solar Abs.[0.0 - 0.99])
1 AGED (D)
2 NEW

Par 3 = IWET	(WETNESS CONDITION [1 - 3]) 1 COVERED (DRY) **2 EXPOSED (NORMAL) (D)** 3 WET
Par 4 = N/A	(USE ANY DUMMY VALUE)
Par 5 = N/A	(USE ANY DUMMY VALUE)

DEFAULT CHOICES MARKED WITH (D)

A sample SDF file is illustrated below.

```
30.5 86.5 0.0 1995 06 25 140000 010000 8.0 12.0 1 1
1 8 0 0 1
f4 185 0.0 1 1 1 1
```

Sample SDF File

3.2 Description of the Nodal Parameter File (PAR)

The PAR file is an ASCII file that can be created and edited with a conventional text editor.

The PAR File Format: The nodal parameter file consists of the following groups of input parameter that describes the nodal network for the target model:

A. PARAMETER INPUT QUANTITY TOTALS

A.1 READ nNodes, nConstTempNodes, nRadNodes, nInitTempNodes, nQcurves
A.2 READ nNonRadNodes, nTcurves, nNatConvCards, nForConvCards

nNodes	Total number of nodes (including constant temp & rad nodes)
nConstTempNodes	Number of constant temperature nodes
nRadNodes	Number of radiation nodes
InitTempNodes	Number of nodes requiring individual starting temps and/or power (excluding constant temperature nodes)
nQcurves	Number of time dependent heat rate curves
nNonRadNodes	Number of non-radiation conductors
nTcurves	Number of curves for temp. or time dependent properties (used for conductivity, capacitance, and constant temp nodes)
nNatConvCards	Number of natural conv. cards
nForConvCards	Number of forced conv. cards

B. CONSTANT TEMPERATURES , CURVES, HEAT RATES

B.1 READ ConstTempNodeID, Temperature, ConstTempCurve (1 to nConstTempNodes)

ConstTempNodeID	Constant Temperature Node Number
Temperature	Fixed Temp. with no Curve or Arbitrary with Curve (C)
ConstTempCurve	Curve Number [0 implies no curve]

B.2 READ InitTempNodeID, Temperature, QImposed (1:nInitTempNodes)

InitTempNodeID	Constant Heat Rate (or Individual Start Temp) Node Number
Temperature	Initial Start Temp (other than default Air Temp) (C)
QImposed	Constant Heat Rate (W)

B.3 READ TCurveNodeID, nTCurvePair
B.4 READ X, Y (1:nTCurvePair)
(Repeat Above Two Lines from 1 to nTcurves)

TCurveNodeID	Curve Number
nTCurvePair	Number of Pairs of X,Y Values
X	Time or Temperature (HHMMSS or C)
Y	Temperature or Multiplier

[In time-temperature mode for a conductor curve, a -999 will produce a zero conductance]

B.5 READ QCurveNodeID, nQCurvePair
B.6 READ Time, HeatRate (1:nQCurvePair)
(Repeat Above Two Lines from 1 to nQcurves)
[Arranged in Order of Ascending Node Numbers]

QCurveNodeID	Time Dependent Heat Rate Node Number
nQCurvePair	Number of Time vs. Heat Rate Data Pairs
Time	Dependent Time Value (HHMMSS)
HeatRate	Independent Heat Rate Value (W)

C. CAPACITANCE INPUT

C.1 READ SinCap, StrCap

SinCap	Number of Input Lines of Single Entry Nodal Capacitance
StrCap	Number of Input Lines of Continuous Node Strings of Identical Nodal Capacitance

C.2 READ NodeID, Capacitance, CapacitCurve (Repeat from 1 to SinCap)

NodeID	Node Number
Capacitance	Thermal Capacitance (J/C) [Density X Specific Heat X Area X Thickness]
CapacitCurve	Curve Number of Temperature Dependent Multiplier Value [Usually Set to 0]

C.3 READ StartNodeID, EndNodeID, Capacitance, CapacitCurve
(Repeat from 1 to StrCap)

StartNodeID, EndNodeID	Consecutive Node Numbers From and Including StartNodeID through EndNodeID

D. CONDUCTOR INPUT

NOTE 1: Surface to Surface, BKG to Surface, and SKY to Surface Multiple Bounce Radiation connections are specified through the LWX file (Section 3.3).

NOTE 2: Mass Transfer (Evaporation/Condensation), Precipitation, and Solar Load are automatically applied. The solar absorptivity value (SolarAbsorptivity) will influence the total solar irradiation absorbed by the surface. The fractional interface area (FractionalInterfaceArea) will influence the amount of surface area participating in mass transfer.

NOTE 3: The Wind Convection conductors must be included directly in the PAR file if they are an environmental effect. The characteristic dimension (SignificantDim) will inversely affect the magnitude of the wind heat transfer coefficient (ForcedConvMode 7).

D.1 READ StartNodeID, EndNodeID, Conductance, ConductType (1 to nNonRadNodes)

StartNodeID, EndNodeID	Pair of Node Numbers Linked Through Heat Transfer
Conductance	Effective Conductance or Multiplier Value [Includes Conduction, Convection, Radiation, Fluid Flow]
ConductType	Conductor Type
for ConductType -1	Conductance = Radiation Conductor (M^2) Conductance= $\mathscr{F}_{12}$ X Area_1 or $\mathscr{F}_{21}$ X Area_2
for ConductType 0	Conductance = Constant Conductance or Convection (W/C) Conductance= Conductivity X Area/Length or Conductance= Heat Transfer Coefficient X Area

for ConductType 1 - 100	Two Choices: 1) Time Dependent - Provide a Constant Temp. Node with a Curve Number, Use this Curve Number for ConductType. Conductance value calculated as above. In the Time vs Temp Curve above, a Value of -999 is used when you want Conductance = 0 (off) and Actual Temp. Values when you want Conductance = Conductance (on). 2) Temperature Dependent - Provide a Curve (use as ConductType) of Temperature Dependent Conductivity Values (or Multiplier) in curves above. Conductance = Area/Length (or equivalent conductance with above multiplier)
for ConductType 101 - 200	Natural Convection Conductor Conductance = Area (M^2) Value of (ConductType - 100) References the Record (or Line) in the Natural Convection Library Parameters of Category F.1 Below.
for ConductType 201 - 300	Forced Convection Conductor Conductance = Area (M^2) Value of (ConductType - 200) References the Record (or Line) in the Forced Convection Library Parameters of Category F.2 Below. ConductType = 201 is Special Case Reserved for Wind Convection.
for ConductType 301 - 400	Fluid Conductor Conductance = Mass Flow Rate X Specific Heat (W/C) **[Note: Order is Critical, NA is the Downstream Node & NB is the Upstream Node]** If ConductType = 301; Conductance = Constant Value If ConductType> 301; Conductance = Constant Value X Time Curve Value where Curve Value is interpolated from Curve # = ConductType - 300 **[Note: The curves are above and can be any time dependent curve except #1. It is not necessary to set up a Constant Temp Node with this Curve # as required for ConductType of 1 - 100.]**

E. INPUT VALUES OF RADIATION NODES

E.1 READ RadNodeID, Area, ThermalEmissivity, SolarAbsorptivity, SurfaceNodeToSkyViewFactor, SurfaceNodeToEarthViewFactor, FractionalInterfaceArea, NormalVector[3], GroupNumber
(Repeat 1 to nRadNodes)

RadNodeID	Node Number: Positive for Target and Negative for Other
Area	Area [M^2]
ThermalEmissivity	[-]
SolarAbsorptivity	[-]
SurfaceNodeToSkyViewFactor	[-]
SurfaceNodeToEarthViewFactor	[-]
FractionalInterfaceArea	Participating in Mass Transfer
NormalVector[3]	Unit Normal Vector
GroupNumber	Group Number of Node (optional)

F. CONVECTION LIBRARY PARAMETERS

F.1 READ NatConvMode, SignificantDim
(Repeat 1 to nNatConvCards)

NatConvMode	Denotes Specific Natural Convection Mode
SignificantDim	Significant Dimension (M)

(See below for additional detail on natural convection)

F.2 READ ForcedConvMode, Velocity, HydraulicDiameter, SignificantDim, LeadingEdgeNode
[Must Include at Least 1 for ConductType=201 Wind]
(Repeat 1 to nForConvCards)

ForcedConvMode	Denotes Specific Forced Convection Mode
Velocity	Velocity (M/S) or Arbitrary
HydraulicDiameter	Hydraulic Diameter (M) or Arbitrary
SignificantDim	Significant Dimension (M)
LeadingEdgeNode	Arbitrary

(See below for additional detail on forced convection)

G. AERODYNAMIC HEATING PARAMETERS

G.1 READ AeroNode, xLocation,BladeThickness, RadialLocation, AngularSpeed,CurveID
(Repeat as required)

AeroNode	Surface Node Participating in Aero Heating
xLocation	X Location of Node from Leading Edge (M)
BladeThickness	Blade Thickness (M)
RadialLocation	Radial Location of Node from Hub (M)
AngularSpeed	Angular Speed of Rotor (RPM)
CurveID	Curve Number for Angular Speed Multiplier

NOTES: AeroNode is designated by the surface node number. CurveID is treated like other time curve applications in TCM2. If no curve is desired, then give 0 as the curve number and no multiplier will be applied to AngularSpeed. If a positive curve number is used, then a time dependent multiplier will be applied. The same or unique curve numbers can be used throughout the aero nodes. The standard format for designating curves (X,Y) is as indicated above.

Table 3: Additional Detail for Category F.1 — Natural Convection

Mode of Heat Transfer	NatConvMode	SignificantDim
Vertical plate or cylinder	1	Plate height *H*, cylinder height *H*
Horizontal rectangular plate Convection from *upper* surface to air ($T_s > T_{AIR}$) or convection from air to *lower* surface ($T_{AIR} > T_s$).	2	$\frac{WL}{[(W+L)2]}$
Horizontal rectangular plate Convection from air to *upper* surface ($T_{AIR} > T_s$) convection from *lower* surface to air ($T_s > T_{AIR}$).	3	$\frac{WL}{[(W+L)2]}$
Horizontal air space Heat transfer in upward direction. *b* = air space thickness. One node required at midspace.	4	*b*
Vertical air space Heat transfer in horizontal direction. *b* = air space thickness. One node required at midspace.	5	*b*
Small rectangular plate Vertical orientation, $H \lesssim 6$ in. Heat transfer to or from either surface	6	*H*
Horizontal orientation *H, L* ≤ 6 in. Convection from *upper* surface to air or convection from air to *lower* surface.	7	$\frac{WL}{[(W+L)2]}$
Convection from air to *upper* surface or convection from *lower* surface to air.	8	$\frac{WL}{[(W+L)2]}$
Shield internal convection, L < 13mm Heat transfer is from plate surface to plate surface	9	Plate Separation L, Air gap L

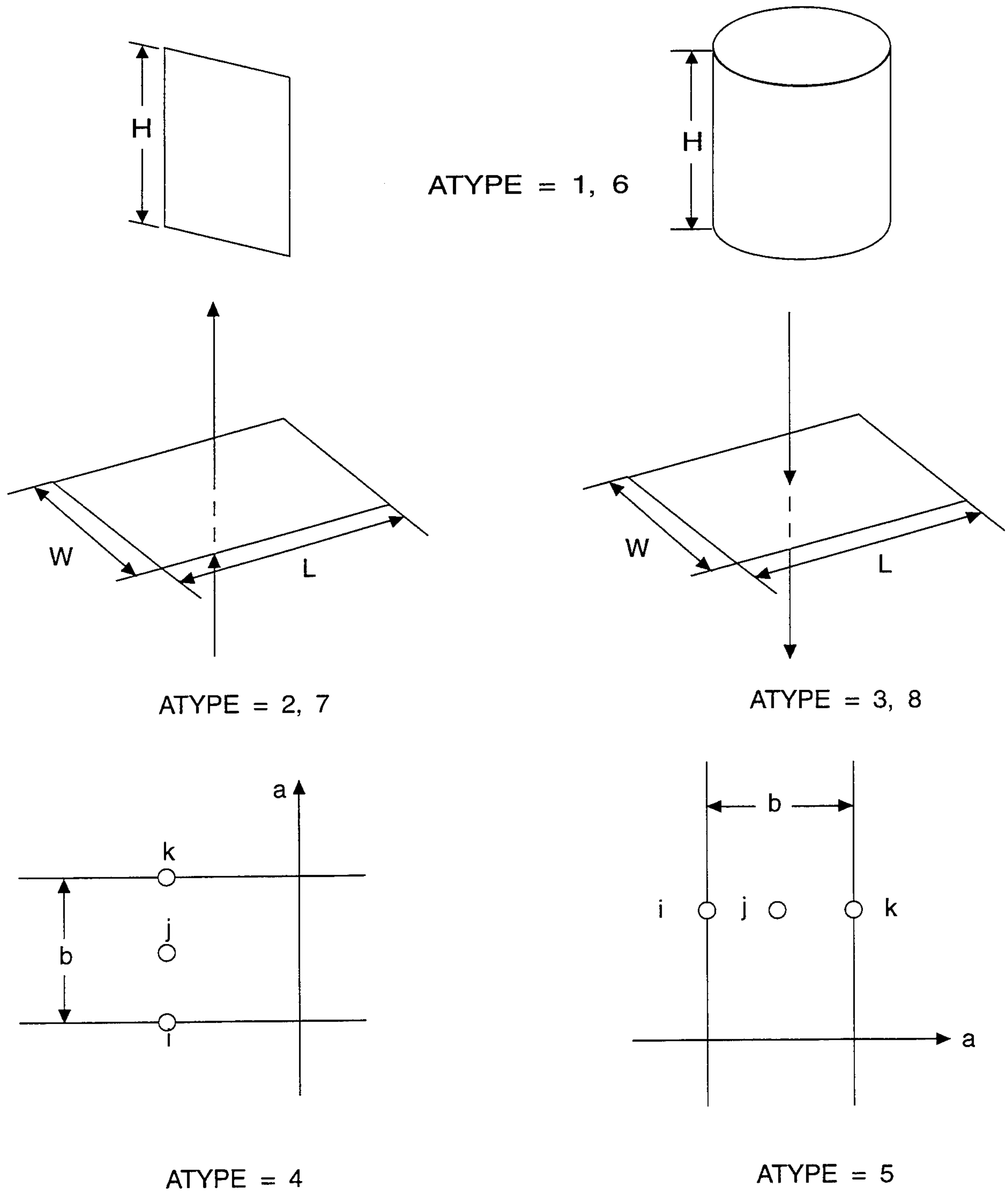


Figure 1: Free Convection Geometry

Table 4: Additional Detail for Category F.2 — Forced Convection

Mode of Heat Transfer	Mode*	V**	D_H	L_c	Node
Duct Flow, (laminar or turbulent)	1	V	D_H	L	A
Duct Flow, (turbulent only) ($Re_D > 10{,}000$)	2	V	D_H	L	A
Flat Plate Flow, (laminar or turbulent)	3	V	A	L	A
Flat Plate Flow, (turbulent only) ($Re_L > 5 \times 10^5$)	4	V	A	L	A
Future Expansion	~~5~~				
Future Expansion	~~6~~				
Wind Convection	7	A	A	$\sqrt[3]{LWH}$	A
Simple Convection Formula $h = C2 + (C1 * V)$	8	V	C1	C2	A

A: Arbitrary, but required numeric input
ΔX: Length of node in string of equal length nodes [M]
L: Length of plate or duct [M]
W: Width of plate or duct [M]
H: Height of plate or duct [M]
V: Flow velocity [M/S]
D_H: Hydraulic diameter [M]
C1,C2: Constants: C2 and C1*V must have same units as h [W/M^2C]

* A negative number will use the time dependent relative airspeed of the wind and vehicle for the velocity, V.

** A negative number denotes a curve number to use from the model.in file which provides velocity, V, as a function of time.

Curve Options: The uses of curve options include: switch conductors on/off (time), vary constant temperature values (time), vary conductances (time or temperature), vary capacitances (temperature), vary aero heating angular velocity (time), and vary emissivity (temperature). All of these controls are handled in the PAR file.

All time dependent quantities **except for aero heating** will require a constant temperature node with the curve number to be set up in the PAR file whether it is directly used or not. This is how TCM2 can tell that it should be interpolating as a function of time instead of temperature. A two step process is used for the time dependent curves: first the constant temperature node is given its new temperature value for that time from the curve data and then conductors that use the same curve number are switched on or off by checking to see if the temperature flag is off (-999) or on (realistic value). This option will only switch the conductor on or off and therefore cannot be used as a multiplier too. The conductor can be connected with any two valid node numbers and does not require that the constant temperature with the same curve number be used. If only the constant temperature variations with time are desired with no conductor switches, then it is not necessary to use the curve number with any of the conductors.

If multipliers for either conductance or capacitance are desired, than these should be used as a function of temperature. **Do not use the same curve number with a constant temperature curve number**. If the conductances vary with temperature, the average of the two connecting nodes is used to provide a new multiplier. In the case of capacitances, the multiplier is based on node temperature. Note that the value obtained by the curve is **multiplied** by the initial entry for capacitance/conductance in the PAR file. If a constant multiplying factor (not as a function of temperature or time) is desired for purposes of varying one or several capacitances/conductances from the IN file, then simply use two X,Y pairs that will not allow any variation with temperature: -10000. factor 10000 factor.

A sample PAR file is illustrated below.

```
   180     5    43     0     0 Residential Construction
   216     1     0     1   RG GRADE GENERIC BUILDING
    42     0.00    0  BACKGROUND
    43     0.00    0  SKY
    44     0.00    0  AIR
   178    68.00    1  Air-Conditioned
   179    69.00    0
1 2                           ; Curve 1
00000 20.0 240000 20.0        ; Controlled Interior Temp ( AC or Heat )
    174     0   SINCAP STRCAP
     1      0.1350E+06   0    1-LVL 1-WALL  1-SCT  WALL SURFACE NODE
     2      0.9000E+05   0    1-LVL 2-WALL  1-SCT  WALL SURFACE NODE
     .
     .
   176      0.1016E+05   0    3-LVL 4-WALL  1-SCT  4-SubLayer
   177      0.3429E+07   0    1-SEG  1-SCT  1-SubLayer (ROOF)
     1      45    215.3      0    1-LVL 1-WALL  1-SCT
    45      46    55.24      0
    46      47    55.86      0
    47      48    212.2      0
    48     178    124.3      0
     .
     .
    38      44    10.00      201
    39      44    15.00      201
    40      44    10.00      201
    41      44    150.0      201
    44     180     1.0E+30     0   WIND TO MASS-FLOW DUMMY
   1    15.00      0.9300  0.1600  0.5000  0.5000 1.0  1.000  0.000  0.000  1
   2    10.00      0.9300  0.1600  0.5000  0.5000 1.0  0.000 -1.000  0.000  2
   .
   .
  40    10.00      0.9300  0.1600  0.5000  0.5000 1.0  0.000  1.000  0.000  4
  41    150.0      0.9000  0.7400  1.0000  0.0000 1.0  0.000  0.000  1.000  5
 -42    1.000      1.0000  0.8000  1.0000  0.0000 1.0  0.000  0.000  1.000  0
 -43    1.000      1.0000  0.0000  0.0000  0.0000 0.0  0.000  0.000  0.000  0
     7  0  0     6.43      0   WIND LIBRARY ELEMENT
```

Figure 2: Sample PAR File

3.3 Description of the Radiation Exchange File (LWX)

The longwave radiation which arrives at a surface node after being reflected (perhaps several times) from other surface nodes is calculated using information contained in the LWX file. This information is in the form of longwave radiation exchange (B_{ij}) factors which are defined as the fraction of radiation in the infrared part of the spectrum that leaves node i and eventually arrives at node j. B_{ij} factors for each surface node are computed from view factor information by the program ScriptF or by RadX.

The LWX File Format: The organization of the LWX file is as follows.

Title Line CRC Number

Header Line specifying the number of surface nodes for which there is longwave radiation exchange information

Header Line specifying the Terrain Emissivity

For each surface node
 Blank Line

 Header Line: "Region Area Emissivity"

 Node number, area, and emissivity

 Header Line specifying the number of surface nodes (NF) plus ground plus sky that exchange radiation with this node

 NF pairs of numbers (an integer and a real) specifying a node number and the Bij to that node

 Line containing "Gnd" followed by the Bij to the ground and "Sky" followed by the Bij to the sky

A sample LWX file is illustrated below.

```
Longwave Radiation Exchange Factors for m2
Number of Exterior Regions = 60
TEMIS     0.940

Region     Area     Emissivity
    1     0.7270     0.9400
Bij       12
    1     0.0020    9     0.1251     16   0.0154     17   0.0248     63   0.0168
   64     0.0107   65     0.0155     66   0.0108     67   0.0082     68   0.0110
  Gnd     0.0013  Sky     0.7421

Region    Area     Emissivity
    3     0.9292     0.9400
Bij 6
    9     0.1779     16   0.0237     17   0.0115     52   0.0053
  Gnd     0.0013      Sky 0.7769
```

Figure 3: Sample LWX File

3.4 Description of the Apparent Area File (APA)

TCM2 uses a table of apparent surface node areas for given sun azimuths and elevations to compute direct solar loading. A distinction is made between apparent areas which account for the effects of shadowing and projected areas which are simply the dot product of the surface normal with the unit vector pointing to the sun. The information for this table is contained in the APA or "Apparent Area" file (model.apa).

The APA File Format: The organization of the APA file is as follows.

> Title Line
> Header Line specifying the number of surface nodes for which there is apparent area information
>
> For azimuth = 0 to 360 in increments of 10 degrees
> Blank Line
> Header Line specifying the current solar azimuth
> Header Line: "Region Area"
>
> For each surface node
> Node number and node area
>
> 10 real numbers specifying the apparent area (as a fraction of the node area) for the current solar azimuth and for solar elevations 0 to 90 in increments of 10 degrees

A sample APA file is illustrated below.

```
Apparent Areas for m2
Number of Exterior Regions = 60

Azimuth =       0.00
Region          Area
        1        0.7276
0.0018  0.0927   0.2091  0.4263  0.6190  0.7382  0.8349  0.9063  0.9502  0.9652
        3        0.9290
0.0000  0.1667   0.3320  0.4799  0.616   0.7352  0.8312  0.9019  0.9452  0.9597
        4        0.8913
0.0000  0.0000   0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
        5        0.8912
0.0000  0.0000   0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
        6        2.8760
0.0830  0.1801   0.2675  0.3699  0.4571  0.5324  0.6273  0.6667  0.6906  0.7073
        7        0.9642
0.0012  0.0164   0.0604  0.1275  0.1634  0.1945  0.2196  0.2380  0.2492  0.2528
        8        0.8475
0.0000  0.0000   0.0000  0.0000  0.0000  0.0000  0.0451  0.0922  0.1364  0.1766

Azimuth = 10.00
Region Area
1       0.7276
0.0277  0.1266   0.2285  0.4953  0.6528  0.7665  0.8570  0.9214  0.9578  0.9652
        3        0.9290
0.0000  0.1667   0.1795  0.4909  0.6345  0.7588  0.8600  0.9351  0.9452  0.9597
        4        0.8913
0.0181  0.0279   0.0456  0.0274  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
        5        0.8912
0.0148  0.0434   0.0244  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
        6        2.8760
0.0881  0.1846   0.2610  0.3633  0.4604  0.5249  0.6226  0.6677  0.6911  0.7073
        7        0.9642
0.0264  0.0804   0.1570  0.1914  0.2200  0.2419  0.2565  0.2632  0.2620  0.2528
        8        0.8475
0.0000  0.0000   0.0000  0.0000  0.0000  0.0229  0.0655  0.1061  0.1435  0.1766
```

Figure 4: Sample APA File

3.5 Description of the Facet File (FAC)

The facet file (model.fac) is created by the thermal modeler and contains target geometry information.

The FAC File Format: The facet file contains the boundary boxes for a target and coordinates of each facet vertex and unit normal. It contains the following types and order of records.

RECORD	LENGTH (WORDS)	CONTENTS
1	6	Xmin,Xmax,Ymin,Ymax,Zmin,Zmax
:	:	[9 blank lines]
10	6	
11	1	'MODEL'
12	2	MID,NF
13	8	FacetID, FacetClass, Vertex1[3], Vertex2[2]
14	8	FacetID, FacetGroup, Vertex3[3], Normal[3]
		Repeat 13,14 I=1,NF)

Xmin..	Bounding Box
MID	Model ID (usually 1)
NF	Total Number of Facets
FacetID	Facet Number (ascending order)
FacetClass	Facet Class (Thermal Surface Node) (nonsequential is allowable)
Vertex	Facet Vertex Coordinates (X_i,Y_i,Z_i)
FacetGroup	Facet Group (Target Component) (Positive Nonzero only; use 1 if not Grouped)
Normal	Unit Normal Coordinates

Vertex Order:	Counter Clockwise
Units:	Meters
Coordinate System:	Positive X = North, Positive Y = West, Positive Z = Vertical
Comments:	User comment lines can be placed on lines in between facets by using the FORTRAN convention of placing "C" in the first column of the comment line (any number of lines can be used). Do not start comments until after line 12 and do not put comments in between two lines describing a facet. Typical use would be to place a comment line prior to each "class" of facets as well as between each "group" of facets.

A sample FAC file is listed below.

```
   -3.5500     3.5500    -1.4000     1.4000     0.0000     2.5330
      {blank}
      {blank}
      {blank}
      {blank}
      {blank}
      {blank}
      {blank}
      {blank}
      {blank}
MODEL
  1   453
C  BED.LFT.W
    1    1     0.4500     1.2250     1.3830    -3.5500     1.2250     1.3830
    1    1    -3.5500     1.2250     1.8330     0.0000     1.0000     0.0000
    2    1     0.4500     1.2250     1.3830    -3.5500     1.2250     1.8330
    2    1     0.4500     1.2250     1.8330     0.0000     1.0000     0.0000
C  BED.LFT.C
    3    2     0.4500     1.2250     1.8330    -3.5500     1.2250     1.8330
    3    1    -3.5500     1.2250     2.3830     0.0000     1.0000     0.0000
    4    2     0.4500     1.2250     1.8330    -3.5500     1.2250     2.3830
    4    1     0.4500     1.2250     2.3830     0.0000     1.0000     0.0000
    :    :          :          :          :          :          :          :
C  W.TT.R.1
  441   57    -1.5750    -1.4000     0.0000    -1.5750    -1.0000     0.0000
  441    1    -1.0750    -1.0000     0.0000     0.0000     0.0000    -1.0000
  442   57    -1.5750    -1.4000     0.0000    -1.0750    -1.0000     0.0000
  442    1    -1.0750    -1.4000     0.0000     0.0000     0.0000    -1.0000
  443   57    -1.0750    -1.4000     0.0000    -1.0750    -1.0000     0.0000
  443    1    -0.8250    -1.0000     0.4330     0.8660     0.0000    -0.5000
  444   57    -1.0750    -1.4000     0.0000    -0.8250    -1.0000     0.4330
  444    1    -0.8250    -1.4000     0.4330     0.8660     0.0000    -0.5000
  445   57    -3.1000    -1.4000     0.4330    -3.1000    -1.0000     0.4330
  445    1    -2.8500    -1.0000     0.0000    -0.8660     0.0000    -0.5000
  446   57    -3.1000    -1.4000     0.4330    -2.8500    -1.0000     0.0000
  446    1    -2.8500    -1.4000     0.0000    -0.8660     0.0000    -0.5000
  447   57    -2.8500    -1.4000     0.0000    -2.8500    -1.0000     0.0000
  447    1    -2.3500    -1.0000     0.0000     0.0000     0.0000    -1.0000
  448   57    -2.8500    -1.4000     0.0000    -2.3500    -1.0000     0.0000
  448    1    -2.3500    -1.4000     0.0000     0.0000     0.0000    -1.0000
  449   57    -2.3500    -1.4000     0.0000    -2.3500    -1.0000     0.0000
  449    1    -2.1000    -1.0000     0.4330     0.8660     0.0000    -0.5000
  450   57    -2.3500    -1.4000     0.0000    -2.1000    -1.0000     0.4330
  450    1    -2.1000    -1.4000     0.4330     0.8660     0.0000    -0.5000
C  W.HO.R.1
  451   58     2.9500    -1.4000     0.4330     2.5750    -1.4000     0.6495
  451    1     2.5750    -1.4000     0.2165     0.0000    -1.0000     0.0000
  452   58    -1.0750    -1.4000     0.4330    -1.4500    -1.4000     0.6495
  452    1    -1.4500    -1.4000     0.2165     0.0000    -1.0000     0.0000
  453   58    -2.3500    -1.4000     0.4330    -2.7250    -1.4000     0.6495
  453    1    -2.7250    -1.4000     0.2165     0.0000    -1.0000     0.0000
```

Figure 5: Sample FAC File

3.6 Description of the Weather File (NC)

The weather and environmental data needed by TCM2 is extracted from weather files (name.nc) with a sponsor-defined format. The weather data must be stored in Network Common Data Form (netCDF) and is accessed from TCM2 using netCDF interface library functions made available by the Unidata Program Center of the University Corporation for Atmospheric Research. The subset of the data contained in these weather files that is required by TCM2 is described below. The description is in network Common Data form Language (CDL). Two important modifications/exceptions are taken to the CDL definition below.

1) The **pal** field should contain the air pressure in millibars and not "IR thermopile W/m^2 using old coefficient".

2) The **water_amt** field value is expected to contain the rainfall rate in inches per minute and not in millimeters per minute.

The netCDF Weather File Format:

```
netcdf min_wea {
dimensions:
        time = UNLIMITED ;
        comment_lines = 1000 ;
        comment_columns = 80 ;
        one = 1 ;
        small_string = 4 ;
        tpq11_range = 256 ;
        num_pp_samples = 16 ;
        prec_string_size = 8 ;

variables:
        long time(time) ;
             time:units = "seconds since Jan 1, 1970 12 Midnight" ;
        float solar_zenith(time) ;
             solar_zenith:missing_value = 99999. ;
             solar_zenith:long_name = "Solar zenith angle relative to target zenith" ;
        float solar_azimuth(time) ;
             solar_azimuth:missing_value = 99999. ;
             solar_azimuth:long_name = "Solar azimuth angle relative" ;
        long wd1(time) ;
             wd1:long_name = "wind direction instantaneous" ;
             wd1:units = "degrees" ;
             wd1:valid_range = "0,360" ;
             wd1:missing_value = 99999 ;
```

```
float ws1(time) ;
      ws1:long_name = "wind speed 1 minute maximum" ;
      ws1:units = "m/s" ;
      ws1:valid_range = "0,100" ;
      ws1:missing_value = 99999 ;
float ta1(time) ;
      ta1:long_name = "air temperature" ;
      ta1:units = "degrees C" ;
      ta1:valid_range = "-20.0,250.0" ;
      ta1:missing_value = 99999 ;
long rh1(time) ;
      rh1:long_name = "relative humidity" ;
      rh1:units = "%" ;
      rh1:valid_range = "0,100" ;
      rh1:missing_value = 99999 ;
float dp1(time) ;
      dp1:long_name = "dew point temperature" ;
      dp1:units = "degrees C" ;
      dp1:valid_range = "-20.0,250.0" ;
      dp1:missing_value = 99999 ;
float sr1(time) ;
      sr1:long_name = "solar radiation diffuse downwelling  0.28-2.8um" ;
      sr1:units = "W/m^2" ;
      sr1:valid_range = "0,1500" ;
      sr1:missing_value = 99999 ;
float sr2(time) ;
      sr2:long_name = "solar radiation total downwelling 0.28-2.8um" ;
      sr2:units = "W/m^2" ;
      sr2:valid_range = "0,1500" ;
      sr2:missing_value = 99999 ;
float ir1(time) ;
      ir1:long_name = "IR radiation upwelling 2.8 - 50um" ;
      ir1:units = "W/m^2" ;
      ir1:valid_range = "0,1500" ;
      ir1:missing_value = 99999 ;
float ir2(time) ;
      ir2:long_name = "solar radiation downwelling 2.8 - 50um" ;
      ir2:units = "W/m^2" ;
      ir2:valid_range = "0,1500" ;
      ir2:missing_value = 99999 ;
```

```
    float pa1(time) ;
        pa1:long_name = "IR thermopile W/m^2 using old coefficient" ;
        pa1:units = "W/m^2" ;
        pa1:valid_range = "0,2000" ;
        pa1:missing_value = 99999 ;
    float water_amt(time) ;
        water_amt:long_name = "water amount in last 1 min period" ;
        water_amt:units = "mm" ;
        water_amt:missing_value = 99999. ;
}
```

All 14 of these fields must be defined as shown above and contain valid data over the simulation period. In addition, if a distinct weather averaging interval is specified in the options file, then the weather file should contain valid values for each of these 14 variables over the entire averaging interval. This requirement can be relaxed for two of the solar and two of the infrared fields, sr1, sr2, ir1, and ir2. If data is not available for these fields, the missing value (**99999**) can be encoded in any of these four fields. If either or both sr1 and sr2 are recorded as missing, TCM2 will call the Insol subroutine. This subroutine implements a simple three layer atmospheric model developed by Hughes STX to compute direct and diffuse insolation. If either or both ir1 and ir2 are recorded as missing, then the SkyRad subroutine is called to predict the effective thermal and infrared temperature of the sky. This function implements STX's modification to Idso's empirical formula for the effective emissivity of the sky. Currently the subroutines are called assuming clear sky conditions.

3.7 Description of the Numeric Parameter File (PRM)

The numerics file (numeric.prm) contains information for the thermal model numerical solving routines (convergence criteria, relaxation factors, etc.).

The PRM File Format: The numeric parameters file contains the single record described below.

READ DTME, NLOOP, BETA, SALDT, TALDT, DELT, LOOPEN

DTME	Transient Time Step (Seconds) [60 - 300]
NLOOP	Number of Numerical Solution Iterations [100 - 2000]
BETA	Over Relaxation Constant [.5 - 1.0 - 1.5]
SALDT	Max SS Allowable Temp Change per Iteration [.001 - .05]
TALDT	Max Transient Allowable Temp Change per Iteration [.001 - .05]
DELT	Transient Max Temp Change per Time Step [25 - 150]
LOOPEN	Number of Iterations between SS Energy Balance [2000 - or less for debug] {future implementation}

Note: The suggested range is for typical cases and is only for guidance. Special cases may warrant exceeding these suggested ranges.

A sample PRM file is illustrated below.

```
180 200 1.0 .05 .05 25 200
```

Sample PRM File

Several run time parameters are accessible from the PRM file to control the speed and accuracy of the individual TCM2 model.

Convergence Criteria: The maximum number of iterations for both steady state and transient calculations is specified with the parameter **NLOOP**. Typically 200 iterations is a sufficient maximum to insure reasonable convergence. Unstable models (i.e., models with highly

varying nodal time constants) may require more iterations to converge during steady state. Often these minor instabilities will "wash" out through the transient execution. If the model does not converge within the maximum number of iterations, it is recommended to extend NLOOP (up to 2000) and then do a "spot" check on resultant temperatures. If temperature differences are insignificant, then it is practical to reduce the NLOOP parameter.

Two related parameters to the NLOOP convergence maximum are the steady state allowable temperature change parameter, **SALDT**, and the same parameter for the transient, **TALDT**. Smaller tolerances will require more iterations (longer run times) but will produce higher accuracy. The improvement in accuracy versus extended run times are the tradeoffs that the analyst must choose. Tolerances of 0.001 are the most stringent that would be recommended and tolerances of 0.05 are the least stringent.

A relaxation factor, **BETA**, is used to accelerate convergence or dampen oscillatory behavior. A BETA of 1.0 would produce a straight Gauss-Seidel procedure with no relaxation. When BETA is less than 1.0, the solution is under relaxed thus dampening oscillations. When BETA is greater than 1.0, the solution is over relaxed thus accelerating the convergence. The absolute value of BETA should range from 0.5 to less than 2.0. This optimum value is usually derived from experimentation and will vary from model to model. It is recommended to use 1.0 and then try values above and below for comparison.

The TCM2 uses an implicit solution method that is unconditionally stable (as compared to the explicit method that requires a stability criterion to be met). Although numerically the solution will not diverge to failure, local oscillations can cause minor temperature fluctuations about the expected value. To minimize this behavior, a maximum transient temperature change parameter, **DELT**, is used. If any node exceeds DELT during a time step, the time step will be appropriately decreased until DELT is no longer exceeded. This is especially effective during cooldowns where the engine is shut off in the middle of a run scenario. It is not particularly effective in making an inherently unstable model behave in a stable manner since most of these instabilities are characterized by subtle oscillations. A DELT of 25°C is typically recommended for most models.

3.8 Description of the Options File

The options file is a text file containing zero or more keyword options. The options that are controlled by this file may be of interest during model development and verification, but are not expected to be generally used since none of these options are needed to predict signatures. The options file should be the third command line argument when TCM2 is invoked

tcm2 sdf-file netCDF-weather-file [options-file]

The options filename is enclosed in brackets to emphasize that it is not required. Do not use the brackets in the actual command line.

The options file can be given any valid filename that does not conflict with another TCM2 input or output filename. The file is composed of blank lines, comments, keywords, and keyword arguments. Comments begin with the symbol ! or # in any column and extend to the end of the line. Keywords are case insensitive, begin with a hyphen, may start in any column. A keyword must be the first non-blank token in a line. Do not include more than one keyword in a single line. Each keyword has a required number of arguments, described below. The arguments are counts, node ranges, dates, and filenames. An argument must not contain the characters !, #, space, or tab. Quoting and escape characters, such as \, are not supported.

The recognized keywords and their arguments are

-Track_Nodes number-of-ranges range-1 ...

This option causes TCM2 to write out the loop count and physical temperatures of the nodes in the requested ranges at the end of each time step. The number-of-ranges argument specifies the number of ranges that follow and must be the same line after the -Track_Nodes keyword. By default the values are written to standard output. A node range is either a single node number or a beginning and ending node number connected by a hyphen. For example,

```
7            # specifies a range composed of the single node 7
11-14        # specifies the range of nodes 11, 12, 13 and 14
```

A hyphenated range must not include any spaces or tabs; for example

```
11 - 14      # Invalid
```

is invalid. The ranges can placed on the line after the keyword and number-of-range argument and on subsequent lines.

-Node_Details

This option produces a detail listing of capacitances, conductances, heat rates, and similar data for all nodes at the completion of the steady state target solution and after the last transient time step for each target. By default the information is written to standard output.

-Optional_OutFile option-output-filename

If an optional output file is designated, any and all energy balance, node tracking, and node detail output will be redirected from standard output to the specified file.

-Weather_Average_Over begin-averaging end-averaging

Averaged weather quantities are, by default, used to compute the initial (steady state) condition of the target. If this option is not invoked, the average of the weather conditions is computed over the simulation interval. An alternate averaging interval is specified by using this keyword. The begin and end averaging dates and times can be on the keyword line or subsequent lines. The format for the dates and times is the same as used in the sdf file, 4 digit year, 1 or 2 digit month, 1 or 2 digit day, 1 to 6 digit integer in the HHMMSS format to specify hours, minutes, and seconds in the range 000000 to 240000. For example,

```
-Weather_Average_Over
    1995 06 25 090000
    1995 06 27 090000
```

The averaging procedure temporarily allocates memory proportional to the number of weather records in the averaging interval, so specifying a long interval can exhaust available computer memory.

-No_Ave_Weather

This keyword suppress computation of average weather conditions.

-No_Solar_Shadowing

Invoking this keyword suppresses use of the solar shadowing data in the APA file. Solar loading of each target surface node is computed using only the orientation of that target surface node and ignores the potential shadowing of the node by other parts of the target.

-ASC_Wea_Dumpfile ascii-weather-filename

Writes the weather data in a text format to the specified file. The format used supports older versions of TCM2 that used a text or FORTRAN unformatted weather file.

-Debugging

Recognized as a valid keyword, but currently has no effect.

-Suppress_Scene_Generate

Recognized as a valid keyword, but currently has no effect.

-Additional_Output

Recognized as a valid keyword, but currently has no effect.

A sample options file is listed below.

```
      # Sample options file

# specifying 4 "ranges" of nodes for tracking
 -track_nodes     4       250-260

      3-6    5                          # second and third ranges

      10-11                                   # fourth range


-optional_OutFile  db.out  # Specifying optional output file named db.out


  -node_details

  -WEATHER_AVERAGE_OVER
      1995 06 23 180000   # begin weather averaging interval
      1995 06 25 180000   # end   weather averaging interval


  -ASC_Wea_DumpFile   ../work/x.xwa
```

Figure 6: Sample Options File

3.9 Description of the Scene Radiance File (SRF)

Contains scene radiances for all targets and backgrounds. This file is read by the PL Scene Builder/Viewer and used for "Thermal Shading".

The SRF File Format: The Scene Radiances File consists of the following types of records.

File Header:	[nT, nB]
Target Header:	[nN]
Node Record:	[nF ǀ FR_1 ǀ ... ǀ FR_{nF}]
Background Record:	[BR]

nT:	integer - number of targets in file
nB:	integer - number of Background Records in file (not counting sky bkg)
nN:	integer - number of thermal nodes in target, i.e. Node Records to follow
nF:	integer - number of facets in the thermal node
FR:	real - facet total radiance (watts/meter^2-steradian)
BR:	real - background total radiance (watts/meter^2-steradian)

File Organization: The Scene Radiances File consists of one File Header, followed by "nT" targets, where a target consists of a Target Header, followed by "nN" Node Records. The last target shall be followed by "nB+1" Background Records, where the first background record contains the sky radiance. The resulting file organization is illustrated below.

```
File Header
        Target Header 1
                Node Record (1,1)
                Node Record (1,2)
                :
                Node Record (1,nN₁)
        Target Header 2
                Node Record (2,1)
                Node Record (2,2)
                :
                Node Record (2,nN₂)
        :
        Target Header nT
                Node Record (nT,1)
                Node Record (nT,2)
                        :
                Node Record (nT,nN_nT)
```

Background Record (sky radiance)
Background Record 1
Background Record 2
:
:
Background Record nB

The order of the targets (i.e., Target Header/Node Record groups) and backgrounds (i.e., Background Record group) correspond to the target and background order in the Scene Description File. The order of the Node Records and facet radiances for each target corresponds to the node and facet ordering in the original FAC file.

A sample SRF file is listed below.

```
1 1
41
2 35.5775 35.5775
2 35.7339 35.7339
.
.
2 35.5046 35.5046
2 36.7761 36.7761
34.830792
36.011791
```

Figure 7: Sample SRF File

3.10 Description of the Apparent Temperature File (APP)

This file contains target and primary background apparent temperatures for the entire simulation.

The APP File Format: The Apparent Temperature File consists of the following types of records.

File Header1:	MxCDA Apparent Temperature File
File Header2:	[Tid, nN, Bid, bkgType, N, E, U]
Time Step Header:	[localTime ǀ airTemperature ǀ extCoeff ǀ tgtBearing]
Target Record:	[$AppT_1$ ǀ $AppT_2$ ǀ $AppT_3$ ǀ ... ǀ $AppT_{nN}$]
Background Record:	[AppBkgT]

Tid:	string - I.D. number for target in file
nN:	integer - number of Nodes in target record
Bid:	integer - I.D. number for primary background in file
bkgType:	integer - Number corresponding to the type of background
N:	The North component of the unit normal to the background
E:	The East component of the unit normal to the background
U:	The Up component of the unit normal to the background
localTime:	real - current time (secs)
airTemperature:	real - current surface air temperature (K)
extCoeff:	real - extinction coefficient for weather conditions at the current time (unitless)
tgtBearing:	real - current target orientation relative to north (deg, pos. towards East)
$AppT_i$:	real - apparent temperature of node i (K)
AppBkgT:	real - apparent temperature of background Bid associated with Tid (K)

File Organization: The file organization is illustrated below.

```
File Header1
File Header2
    Time Step Header 1
        Target Record
        Background Record
    Time Step Header 2
        Target Record
        Background Record
        .
        .
    Time Step Header nTS
```

Target Record
Background Record

Notes: 1) One Target and Background per File, each additional target will be in its own file along with its associated background.
2) The file is read to EOF.

A sample APP file is listed below.

```
MxCDA Apparent Temperature File
frmf 41 1 5 0.000000 0.000000 1.000000
46800 296.22 0.0000 0.00
296.08 296.12 296.07 296.12 296.08 295.20 296.08 295.20 296.08 295.20
296.08 295.20 296.08 296.12 295.21 296.12 295.21 296.12 295.21 296.12
296.07 295.19 296.07 295.19 296.07 295.19 296.07 295.19 296.07 296.12
295.21 296.12 295.21 296.12 295.21 296.12 296.08 296.12 296.07 296.12
298.03
295.96
47100 296.35 0.0000 0.00
296.54 296.79 296.55 296.45 296.54 295.56 296.54 295.56 296.54 295.56
296.54 295.56 296.54 296.79 295.66 296.79 295.66 296.79 295.66 296.79
296.55 295.57 296.55 295.57 296.55 295.57 296.55 295.57 296.55 296.45
295.52 296.45 295.52 296.45 295.52 296.45 296.54 296.79 296.55 296.45
300.01
296.65
.
.
50400 295.65 0.0000 0.00
295.24 295.50 295.29 295.11 295.24 294.46 295.24 294.46 295.24 294.46
295.24 294.46 295.24 295.50 294.52 295.50 294.52 295.50 294.52 295.50
295.29 294.48 295.29 294.48 295.29 294.48 295.29 294.48 295.29 295.11
294.42 295.11 294.42 295.11 294.42 295.11 295.24 295.50 295.29 295.11
297.21
296.01
```

Figure 8: Sample APP File

3.11 Description of the Physical Temperature File (PHT)

The physical temperature file (model.pht) contains target and primary background physical temperatures for the entire simulation. Data in these files can be compared with thermocouple data during model assessment.

The PHT File Format: The format is identical to the previous APP file format with the exception that physical target temperatures are reported rather than apparent target temperatures.

A sample PHT file is listed below.

```
MxCDA Physical Temperature File
frmf 41 1 5 0.000000 0.000000 1.000000
46800 296.22 0.0000 0.00
296.14 296.18 296.12 296.18 296.14 295.19 296.14 295.19 296.14 295.19
296.14 295.19 296.14 296.18 295.20 296.18 295.20 296.18 295.20 296.18
296.12 295.19 296.12 295.19 296.12 295.19 296.12 295.19 296.12 296.18
295.20 296.18 295.20 296.18 295.20 296.18 296.14 296.18 296.12 296.18
298.36
296.01
47100 296.35 0.0000 0.00
296.54 296.80 296.54 296.45 296.54 295.51 296.54 295.51 296.54 295.51
296.54 295.51 296.54 296.80 295.61 296.80 295.61 296.80 295.61 296.80
296.54 295.51 296.54 295.51 296.54 295.51 296.54 295.51 296.54 296.45
295.47 296.45 295.47 296.45 295.47 296.45 296.54 296.80 296.54 296.45
300.38
296.65
.
.
50400 295.65 0.0000 0.00
295.26 295.54 295.33 295.13 295.26 294.44 295.26 294.44 295.26 294.44
295.26 294.44 295.26 295.54 294.51 295.54 294.51 295.54 294.51 295.54
295.33 294.46 295.33 294.46 295.33 294.46 295.33 294.46 295.33 295.13
294.40 295.13 294.40 295.13 294.40 295.13 295.26 295.54 295.33 295.13
297.57
296.12
```

Figure 9: Sample PHT File

4. SLOPED BACKGROUNDS

TCM2 is able to predict the temperature of a background section taking into account its orientation. The orientation of each background section is specified in the SDF file by a surface normal described in the world North-East-Up (*NEU*) coordinate system.

4.1 Assumptions

The influence of the background orientation on the thermal calculations is as follows:

1) Direct solar irradiance is dependent upon the angle of the sun. For this reason, the relative position of the sun with respect to the sloped background is used in calculating direct solar irradiance. It is assumed that the background section is not shadowed by other parts of the background.

2) Diffuse solar irradiance is dependent upon the fractions of the solar and anti-solar regions of the sky which are "seen" by a section of background. TCM2 calculates diffuse solar irradiance on a sloped background surface using equations from Shapiro[4]. These equations use the solar azimuth and zenith angles and the background tilt angle to compute the fractions of the solar and anti-solar regions which are viewed by the sloped surface. The diffuse irradiance on the sloped surface is calculated using these fractions. It is assumed that there is no shadowing of the sloped surface.

3) Reflected solar irradiance is calculated using the total solar irradiance on the surrounding background sections, the view factor from the sloped background surface to the surrounding background sections, and the solar reflectivity of the surrounding background sections. The view factor is calculated assuming that the surrounding background is a horizontal plane. It is also assumed that the solar reflectivity of the surrounding background sections are the same as the solar reflectivity of the sloped background section.

4) Skyshine is dependent upon the view factor from the sloped background surface to the sky. The view factor is calculated using the tilt angle of the background, assuming that the surrounding background is a horizontal plane.

4.2 Thermal Calculations

TCM2 is able to predict the temperature of a vehicle taking into account the bearing of the

[4] Shapiro, Ralph, "A Simple Model for the Calculation of the Flux of Direct and Diffuse Solar Radiation Through the Atmosphere," Scientific Report No. 35, AFGL-TR-87-0200, ST Systems Corp., Lexington, MA, 1987, ADB114709

vehicle (specified in the SDF file as an angle measured CW from north) and that the vehicle may be placed on a sloped background section. Since radiation exchange factors are only available from the vehicle to an infinite background plane, it must be assumed that, although finite, the section of background on which the vehicle is placed is large enough such that radiation exchange with other parts of the background may be ignored. The assumption of a large but finite background section implies that: 1) the underneath the vehicle does not receive direct solar irradiance, and, 2) the slope of the background does not affect the time at which the vehicle receives diffuse solar irradiance. However, the slope of the background does affect the *magnitude* of the diffuse solar irradiance on the vehicle.

The influence of the background orientation and vehicle bearing on the thermal calculations is as follows:

1) Direct solar irradiance is dependent upon the angle of the sun relative to each (vehicle) thermal node. The vehicle's apa file determines the apparent area of each thermal node for 370 solar positions relative to the vehicle. The position of the sun relative to the vehicle is calculated from the orientation of the background, the bearing of the vehicle, and the solar azimuth and zenith angles as measured in the world coordinate system (see below).

2) Diffuse solar irradiance, skyshine, and radiation exchange calculations are dependent upon how much of the sky and background is "seen" from each (vehicle) thermal node. Since the vehicle has the same z axis as the large ground plane on which it is placed, the amount of sky and background that is "seen" from each thermal node is independent of the background orientation. Since the vehicle only "sees" the background section it is on, and since all areas of the same background section have identical properties, the vehicle bearing does not affect any of the radiation exchange factors. In summary, the diffuse solar irradiance of a vehicle on a sloped background is affected only by the change in the fraction of solar and anti-solar regions that are "seen" by a thermal node, the skyshine is independent of the background orientation and vehicle bearing, and the radiation exchange of the vehicle with the background is affected only in that the background itself is influenced by a change in background orientation.

4.3 Position of the Sun Relative to the Vehicle

Coordinate Systems and Transformations: Internally, TCM2 uses a fixed right-handed world coordinate system with principle directions of $\boldsymbol{N}_w$, $\boldsymbol{W}_w$, and $\boldsymbol{U}_w$ (North-West-Up or *NWU*). Each background section has associated with it a coordinate system with unit vectors designated $\boldsymbol{X}_b$, $\boldsymbol{Y}_b$, and $\boldsymbol{Z}_b$, such that $\boldsymbol{Z}_b$ is the direction normal to the background and $\boldsymbol{X}_b$ and $\boldsymbol{Y}_b$ are coincident with $\boldsymbol{N}_w$ and $\boldsymbol{W}_w$ for a non-sloped background section. Each vehicle has associated with it a coordinate system with unit vectors designated $\boldsymbol{X}_v$, $\boldsymbol{Y}_v$, and $\boldsymbol{Z}_v$, such that the positive $\boldsymbol{X}_v$ direction points to the front of the vehicle, positive $\boldsymbol{Y}_v$ points to right side of the vehicle, and positive $\boldsymbol{Z}_v$ points through the top of the vehicle. When a vehicle is placed on a background

section, $\boldsymbol{Z}_v$ is always coincident with $\boldsymbol{Z}_b$.

The transformation of vectors from one coordinate system to another is usually accomplished through the use of a 4×4 matrix known as a *homogenous transform*. Since we are concerned only with the *orientation* of vectors described in one coordinate system relative to another, we need to use only the 3×3 sub-matrix of the homogenous transform which describes rotation. In our notation ${}^{A}\boldsymbol{R}_{B}$ designates the 3×3 *rotation matrix* which gives the components of a vector in coordinate system A of a vector originally described in coordinate system B. In the simple case of converting vectors described in the external NEU coordinate system to NWU vectors, we multiply the E component by negative one, rather than using a 3×3 matrix to accomplish the same thing.

Solution: Given the solar azimuth and zenith angles, the orientation of the background, and the bearing of the vehicle, all of which are described in the external *NEU* coordinate system, this section illustrates the steps involved in determining the solar azimuth and zenith angles relative to the vehicle. We can state our objective as finding the rotation matrix ${}^{v}\boldsymbol{R}_{w}$ such that ${}^{v}\boldsymbol{S} = {}^{v}\boldsymbol{R}_{w} \cdot {}^{w}\boldsymbol{S}$. ${}^{v}\boldsymbol{S}$ is the vector from the origin of the vehicle coordinate system towards the sun, and ${}^{w}\boldsymbol{S}$ is the vector from the origin of the world coordinate system towards the sun, which is determined directly from the solar azimuth angle θ_{az} (measured CW from world north) and solar zenith angle θ_{ze} (measured from world up) angles by

$$
{}^{w}S = \begin{bmatrix} \cos\theta_{az}\sin\theta_{ze} \\ -\sin\theta_{az}\sin\theta_{ze} \\ \cos\theta_{ze} \end{bmatrix} \qquad \textbf{(4-1)}
$$

Placing the vehicle on a sloped background implies that the z direction for the vehicle, $\boldsymbol{Z}_v$, is coincident with the z direction for the background $\boldsymbol{Z}_b$. For this reason, we separate the transform from the world coordinate system to the vehicle coordinate system into two separate transforms, one from the world coordinate system to the background coordinate system, and one from the background coordinate system to the vehicle coordinate system, i.e., ${}^{v}\boldsymbol{R}_{w} = {}^{v}\boldsymbol{R}_{b} \cdot {}^{b}\boldsymbol{R}_{w}$.

We will proceed by first finding ${}^{w}\boldsymbol{R}_{b}$ and then by taking the inverse to get ${}^{b}\boldsymbol{R}_{w}$. ${}^{w}\boldsymbol{R}_{b}$ can be defined by the series of individual rotations required to describe the orientation of the surface normal of the background in the world coordinate system, ${}^{w}\boldsymbol{Z}_{b}$, which in a horizontal background is coincident with ${}^{w}\boldsymbol{U}_{w}$. One possible choice is a rotation by θ_W about the $\boldsymbol{W}_w$ axis (giving the background normal the correct angle from vertical) followed by a rotation by θ_U about the $\boldsymbol{U}_w$ axis (giving the background normal the correct north-west azimuth angle).

We are given the surface normal of the background section, and from that we may find ${}^{w}\boldsymbol{R}_{b}$ by

$$
{}^{w}\boldsymbol{Z}_{b} = \mathbf{ROT}({}^{w}\boldsymbol{W}_{w},\theta_{W}) \cdot \mathbf{ROT}({}^{w}\boldsymbol{U}_{w},\theta_{U}) \cdot {}^{b}\boldsymbol{Z}_{b} = {}^{w}\boldsymbol{R}_{b} \cdot {}^{b}\boldsymbol{Z}_{b} = {}^{w}\boldsymbol{R}_{b} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4-2}
$$

$$
\begin{aligned}
{}^{w}\boldsymbol{R}_{b} = \mathbf{ROT}({}^{w}\boldsymbol{U}_{w},\theta_{U}) \cdot \mathbf{ROT}({}^{w}\boldsymbol{W}_{w},\theta_{W}) &= \begin{bmatrix} \cos\theta_U & -\sin\theta_U & 0 \\ \sin\theta_U & \cos\theta_U & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_W & 0 & \sin\theta_W \\ 0 & 1 & 0 \\ -\sin\theta_W & 0 & \cos\theta_W \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta_U\cos\theta_W & -\sin\theta_U & \cos\theta_U\sin\theta_W \\ \sin\theta_U\cos\theta_W & \cos\theta_U & \sin\theta_U\sin\theta_W \\ -\sin\theta_W & 0 & \cos\theta_W \end{bmatrix}
\end{aligned} \tag{4-3}
$$

solving for θ_W and θ_U:

$$
{}^{w}\boldsymbol{Z}_{b} = {}^{w}\boldsymbol{R}_{b} \cdot {}^{b}\boldsymbol{Z}_{b} \tag{4-4}
$$

$$
\begin{bmatrix} Z_N \\ Z_W \\ Z_U \end{bmatrix} = \begin{bmatrix} \cos\theta_U\cos\theta_W & -\sin\theta_U & \cos\theta_U\sin\theta_W \\ \sin\theta_U\cos\theta_W & \cos\theta_U & \sin\theta_U\sin\theta_W \\ -\sin\theta_W & 0 & \cos\theta_W \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_U\sin\theta_W \\ \sin\theta_U\sin\theta_W \\ \cos\theta_W \end{bmatrix} \tag{4-5}
$$

$$
\theta_U = \mathrm{ArcTan}(Z_N, Z_W) \tag{4-6}
$$

$$
\theta_W = \mathrm{ArcCos}(Z_U) \tag{4-7}
$$

where Z_N, Z_W, and Z_U are the given *NWU* components of ${}^{w}\boldsymbol{Z}_b$ (converted from *NEU*). The two-argument, four-quadrant ArcTan function is used to solve for θ_U since it may take on any value in the range from 0 to 360. ${}^{w}\boldsymbol{R}_b = ({}^{b}\boldsymbol{R}_w)^{-1}$ which is found by substituting the values for θ_W and θ_U into equation 4-3 and then by taking the *transpose* (since ${}^{b}\boldsymbol{R}_w$ has orthonormal columns).

Finding ${}^{b}\boldsymbol{R}_v$ is not quite as straightforward since we are solving for an angle in the background coordinate system given a desired bearing in the world coordinate system. Using β to designate the desired bearing of the vehicle in the *NWU* coordinate system measured CCW from north

(equal to the negative of the bearing specified CW from north by the user) , we know that the north component of the vehicle x axis is proportional to cos β and that the west component of the vehicle x axis is proportional to sin β.

$$ {}^{w}X_{v} = \begin{bmatrix} l\cos\beta \\ l\sin\beta \\ 1-l^{2} \end{bmatrix} \qquad \textbf{(4-8)} $$

Having already solved for θ_W and θ_U, ${}^{w}\boldsymbol{R}_b$ is known. ${}^{b}\boldsymbol{R}_v$ can be defined by the rotation by θ_Z of the vehicle coordinate system about the $\boldsymbol{Z}_b$ axis which gives the vehicle the specified bearing:

$$ {}^{b}\boldsymbol{R}_{v} = \mathbf{ROT}(\boldsymbol{Z}_{b},\theta_{Z}) = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(4-9)} $$

We find the value of θ_Z which gives the specified bearing by solving:

$$ {}^{w}\boldsymbol{X}_{v} = {}^{w}\boldsymbol{R}_{b}\cdot {}^{b}\boldsymbol{R}_{v}\cdot {}^{v}\boldsymbol{X}_{v} \qquad \textbf{(4-10)} $$

$$ \begin{bmatrix} l\cos\beta \\ l\sin\beta \\ 1-l^{2} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_{11}\cos\theta_Z + r_{12}\sin\theta_Z \\ r_{21}\cos\theta_Z + r_{22}\sin\theta_Z \\ r_{31}\cos\theta_Z + r_{32}\sin\theta_Z \end{bmatrix} \qquad \textbf{(4-11)} $$

where the r_{ij} are the elements of ${}^{w}\boldsymbol{R}_b$. The result is:

$$ \theta_Z = \mathrm{ArcTan}\left(\frac{r_{22}\cos\beta - r_{12}\sin\beta}{r_{11}r_{22} - r_{12}r_{21}}, \frac{r_{11}\sin\beta - r_{21}\cos\beta}{r_{11}r_{22} - r_{12}r_{21}} \right) \qquad \textbf{(4-12)} $$

${}^{b}\boldsymbol{R}_v$ is found by substituting θ_Z into equation 4-9. The rotation matrix to transform solar vectors in the *NWU* coordinate system to the vehicle coordinate system, ${}^{v}\boldsymbol{R}_w$, is found by multiplying ${}^{v}\boldsymbol{R}_b = ({}^{b}\boldsymbol{R}_v)^{\mathrm{T}}$ by ${}^{b}\boldsymbol{R}_w$.

Algorithm: The algorithm for finding the solar azimuth and zenith angles in the vehicle

coordinate system is as follows:

1) Convert the solar azimuth and zenith angles to a vector towards the sun in the *NWU* coordinate system.

2) Convert the background surface normal described in the external left-handed *NEU* coordinate system to the internal right-handed *NWU* coordinate system.

3) Use the background surface normal to determine the transform from the background coordinate system to the *NWU* coordinate system.

4) Use the vehicle bearing to determine the rotation of the vehicle relative to the background.

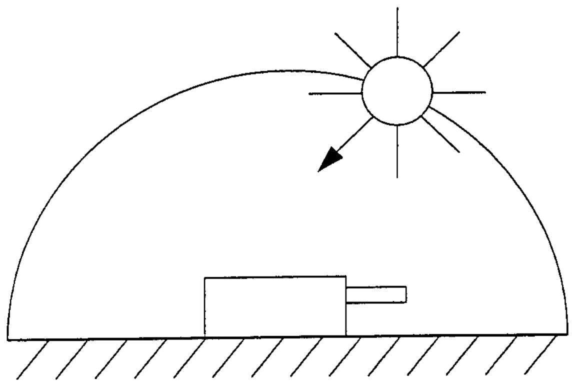
5) Determine the transform from the *NWU* coordinate system to the vehicle coordinate system.

6) Transform the solar vector from the *NWU* coordinate system to the vehicle coordinate system.

7) Convert the solar vector as described in the vehicle system to azimuth and zenith angles.

Solar Components:

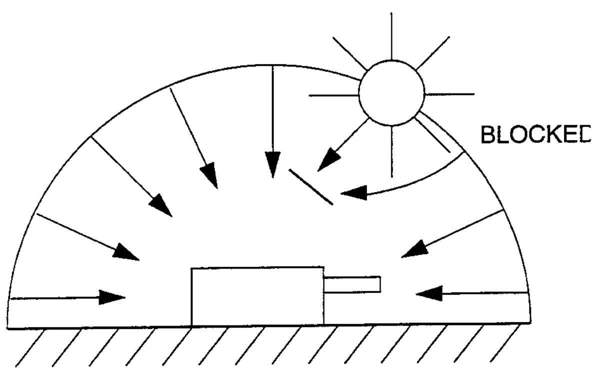
1. Direct or Beam Irradiance = Direct Solar Incidence



(No Diffuse Solar)

Measured by a pyrheliometer (W/m^2)*
(Short Wavelength)

2. Diffuse Solar Irradiance

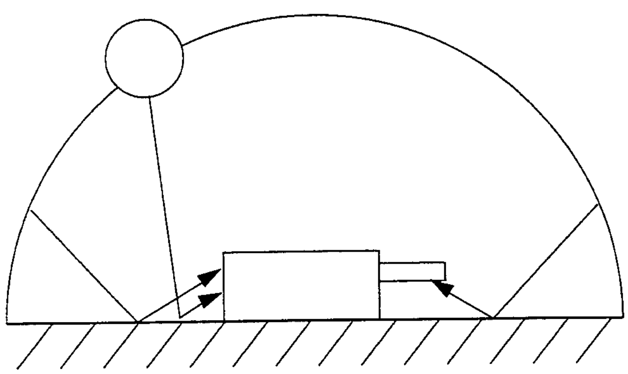


(Not to be confused with thermal skyshine or sky temperature which is long wavelength)

(No direct components)

Measured by a shadow band pyranometer (W/m^2) with correction for diffuse shaded portion (Short Wavelength)

3. Reflected solar or albedo



Total solar (direct + diffuse) reflected from ground

Measured by downlooking pyranometer (W/m^2) **

(Short Wavelength)

Two quantities are either input to TCM2 or calculated from a solar model.

1) Beam — pyrheliometer value
2) Total — pyranometer value

These two quantities can produce the three solar components.

*The beam (direct) can also be obtained with a total pyranometer and a shadow band pyranometer with the following calculation:

$$BEAM = \frac{TOTAL - DIFFUSE}{\cos\theta_Z} \qquad \textbf{(4-13)}$$

where θ_Z = solar zenith angle

**The reflected can also be obtained with the following calculation:

REFLECTED = R_{SOL} TOTAL R_{SOL} = solar reflectivity of ground

5. BRDF AND MODIFIED PHONG SHADING

The BRDF (bi-directional reflectance distribution function) algorithm is described below. It will be used in conjunction with the PL Scene Builder/Viewer and also the TACOM Prism Viewer[5].

The addition of BRDF display capabilities involved work that could be assigned to one of two categories:

1) The implementation of the particular model used to determine waveband dependent directional emission and reflection, given object/observer geometry, and

2) The geometrical relationships between the viewer and the object and between various parts of the object.

In the Sandford-Robertson model[6], the BRDF in a given waveband is characterized by 4 parameters:

ε = hemispherical emittance;
$\rho 1$ = directional-hemispherical diffuse reflectance;
b = geometric parameter governing reflectance at grazing angles; and
e = geometric parameter governing the width of the specular lobe.

The surface BRDF or ρ_{bd} defined as the ratio of the reflected radiance to the incident energy per unit area and time is expressed,

$$\rho_{bd}=\frac{\rho 1}{\pi}\frac{g(\theta_r)g(\theta_i)}{G(b)^2}+\frac{1}{4\pi}\rho_s(\theta_i)\frac{h(\alpha)}{H(\theta_i)}\frac{1}{\cos(\theta_r)} \qquad \textbf{(5-1)}$$

where

$$g(\theta)=\frac{1}{1+b^2\tan^2(\theta)} \qquad \textbf{(5-2)}$$

[5] This task was cost shared between TACOM (PRISM) and PL (TCM2). An initial implementation was done for the PRISM Viewer which displays a BRDF image from a PRISM run. Since the PL Scene Viewer is in progress, the minor tasks of interface files and implementation for BRDF will be done at a later time.

[6] Sandford, Brian P. and Robertson, David C., "Infrared Reflectance Properties of Aircraft Paints," Proceedings of IRIS Targets, Backgrounds and Discrimination, 1985.

$$G(b)=\frac{1}{1-b^2}[1-\frac{b^2}{1-b^2}\ln(\frac{1}{b^2})] \tag{5-3}$$

$$h(\alpha)=\frac{1}{(e^2\cos^2(\alpha)+\sin^2(\alpha))^2} \tag{5-4}$$

$$\cos(\alpha)=\frac{\cos(\theta_i)+\cos(\theta_r)}{\sqrt{2(1+q)}} \tag{5-5}$$

$$q=\cos(\theta_i)\cos(\theta_r)+\sin(\theta_i)\sin(\theta_r)[\cos(\phi_i)\cos(\phi_r)+\sin(\phi_i)\sin(\phi_r)] \tag{5-6}$$

$$\rho_s(\theta_i)=1-(\rho 1+\epsilon)\frac{g(\theta_i)}{G(b)} \tag{5-7}$$

$$\rho(\theta_i)=1-\frac{g(\theta_i)}{G(b)}\epsilon \tag{5-8}$$

Here, θ_i and ϕ_i are the polar and azimuthal angles defining the direction of incidence, θ_r and ϕ_r

$$H(\theta_i)=\frac{1}{2e^2}[(1-e^2)\cos(\theta_i)+\frac{2e^2+(1-e^2)^2\cos^2(\theta_i)}{\sqrt{(1-e^2)^2\cos^2(\theta_i)+4e^2}}] \tag{5-9}$$

are the polar and azimuthal angles defining the reflection direction and α is the angle between the surface normal and the line that bisects the incident and reflected rays[7].

[7] These equations are found in "Software Programmer's Manual - Incorporation of BRDF into IR Signature Analysis" Interim Report, Contract F33615-88-C-1865 of the Electro-Optics Laboratory of the Georgia Institute of Technology, prepared for WRDC/AARI-3.

The directional emittance is given in the Sandford-Robertson model by

$$\epsilon(\theta)=\epsilon\frac{g(\theta)}{G(b)} \tag{5-10}$$

When implementing these expressions in the BRDF signature display, only 'single-bounces' were calculated. That is, the intensity apparent to an observer was based on the directional emission from the point on the object being viewed, and single reflections from all other parts of object that would radiate to the point being viewed, and then be reflected to the observer. Furthermore, the original source of the reflected radiation was taken to be a diffuse emitter, and directionality was accounted for only at the point of reflection - both for the incident ray and the reflected ray.

Since single-bounce reflections were modeled, geometric view-factors were neccessary to calculate inter-object paths of radiation. These are computed at a polygon to polygon level. When the direction dependent reflection of radiation from one polygon off of another and to the observer was computed, polygon-to-polygon view-factors were used to estimate the proportion of the total radiation emitted from the source polygon that reached the reflecting polygon.

Since the BRDF signature was desired on a pixel-level basis, it remained to determine the geometric relations between the observer and the object - at each pixel 'on the object'. When the user selects a given point of view and BRDF signature display, the object image is scanned, with the polygon each pixel 'lies on' identified using pixel level graphics functions. Each pixel is then assigned a unit normal vector to be used in computing the above direction dependent reflectance and emittance.

The object surface is described using a tiling of polygons - with all points on a polygon having a single unit normal vector. In order to display smoothly varying BRDF signatures, smoothly varying unit normals must be assigned to each pixel that lies on a polygon. The approach taken is called Phong shading. In this scheme, the polygon unit normals were averaged to arrive at a unit normal that was assigned to the vertex shared by the polygons. When the BRDF image was to be displayed, each pixel on a horizontal scan-line was examined to determine which polygon it 'belonged to' and where the scan-line it fell on intersected the polygon edges. When these points of intersection were determined, unit normals were interpolated from vertices defining the polygon edges back to the points of intersection. The two interpolated unit normals were in turn used to interpolate a unit normal back at the pixel of interest.

So, the plan was - given a pixel on a particlar scan-line, use a graphics function to find coordinates defining vectors in the same direction as the point-of-view, but at the ends of the scan-line. These two (coplanar) vectors defined a plane that could be intersected with the polygon plane to find the path of the scan-line across the polygon. The intersections of this scan-line path with the polygon edges were computed and unit normal vectors interpolated from the polygon vertices to the points of intersection. Finally, these unit normals were used in interpolating a unit

normal back at the original pixel of interest. With the pixel position and unit normal known, the above Sandford-Robertson equations, along with the BRDF parameters and surface temperatures, could be used to estimate a sum of direction dependent reflected radiances from all the other polygons and a direction dependent emission to the observer.

Some additional BRDF-related modules:

1) ReadExtraDataForBRDF - This module allocates memory for a polygon property data structure. It also fills this structure with polygon normal vectors and centroid postions. It identifies polygon neighbors and saves the information in the 'neighbors' array. It reads the polygon-polygon view factor file and allocates memory for BRDF class assignments - each region is assigned a BRDF class. It reads a PRISM generated file holding region radiances, solar and background radiances and solar position at each display time. It reads in paint data, which holds BRDF parameter information.

2) GetPgonPxlrayInsctn - This routine computes the point of intersection of a line defined by the viewers line-of-sight and the polygon the viewer is looking at.

3) GetPgonEdgeInsctPnts - A horizontal line across the view screen defines a scan line. This routine finds the points of intersection of this scan line with each of the polygons describing the viewed object and at these points of intersection, interpolates a unit normal vector from the normals assigned to the vertices assigned to the polygon edge. This function checks to see if the scan-line intersection with the polygon edges lie between the polygon vertices. It occasionally happens that GetPgonEdgeInsctPnts decides that the scan-line actually passes outside of the polygon that the graphics function claims it lies within. When this disagreement occurs, a failure flag is passed back and neighboring polygons are examined.

4) SRBrdf - This module computes the Sandford-Robertson BRDF reflectance and emittance following the equations given previously.

5) FindBRDFSig - Computes the BRDF radiance for a specified pixel.

6) WhichPolygonsSeeTheSun - If the sun is up, its position and radiance will be in the *.rad file[8]. In order to calculate the reflected solar radiance, we first determine if the polygon is in the sunshine - if so, WhichPolygonsSeeTheSun returns a flag indicating so and the polygon's area as projected in the solar direction is then used to weight the reflected solar radiance.

[8] This is the name of the file in PRISM that contains radiance outputs and future TCM2-BRDF implementation will include a similar interface.

7) FindAllPxlNormals - Examines all pixels forming the object image and uses GetPgonPxlrayInsctn and GetPgonEdgeInsctPnts to assign unit normal vectors to each of the pixels.

8) GetPxlPlaneNormal - Finds a vector normal to the plane formed by two line-of sight vectors obtained from the SGI mapw() function at two points on the scan-line. This is done once for each scan-line.